

Electronic design automation libraries —

Part 1: Input/Output buffer information specifications (IBIS version 3.2)

The European Standard EN 62014-1:2002 has the status of a
British Standard

ICS 25.040

National foreword

This British Standard is the official English language version of EN 62014-1:2002. It is identical with IEC 62014-1:2001.

The UK participation in its preparation was entrusted to Technical Committee GEL/93, Design automation, which has the responsibility to:

- aid enquirers to understand the text;
- present to the responsible international/European committee any enquiries on the interpretation, or proposals for change, and keep the UK interests informed;
- monitor related international and European developments and promulgate them in the UK.

A list of organizations represented on this committee can be obtained on request to its secretary.

Cross-references

The British Standards which implement international or European publications referred to in this document may be found in the BSI Catalogue under the section entitled “International Standards Correspondence Index”, or by using the “Search” facility of the *BSI Electronic Catalogue* or of British Standards Online.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

Compliance with a British Standard does not of itself confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 24 February 2004

Summary of pages

This document comprises a front cover, an inside front cover, the EN title page, pages 2 to 84, an inside back cover and a back cover.

The BSI copyright notice displayed in this document indicates when the document was last issued.

Amendments issued since publication

Amd. No.	Date	Comments

© BSI 24 February 2004

ISBN 0 580 43463 X

Electronic design automation libraries
Part 1: Input/Output buffer information specifications
(IBIS version 3.2)
(IEC 62014-1:2001)

Automatisation de la conception -
Bibliothèques
Partie 1: Spécifications des informations
en entrée/sortie des circuits tampon
(IBIS version 3.2)
(CEI 62014-1:2001)

Bibliotheken für die
Entwurfsautomatisierung
Teil 1: Spezifikation von Eigenschaften
von I/O Buffern
(IBIS Version 3.2)
(IEC 62014-1:2001)

This European Standard was approved by CENELEC on 2001-07-01. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the Central Secretariat or to any CENELEC member.

This European Standard exists in two official versions (English, French). A version in any other language made by translation under the responsibility of a CENELEC member into its own language and notified to the Central Secretariat has the same status as the official versions.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Czech Republic, Denmark, Finland, France, Germany, Greece, Iceland, Ireland, Italy, Luxembourg, Malta, Netherlands, Norway, Portugal, Spain, Sweden, Switzerland and United Kingdom.

CENELEC

European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

Central Secretariat: rue de Stassart 35, B - 1050 Brussels

Foreword

The text of document 93/129/FDIS, future edition 1 of IEC 62014-1, prepared by IEC TC 93, Design automation, was submitted to the IEC-CENELEC parallel vote and was approved by CENELEC as EN 62014-1 on 2001-07-01.

The following dates were fixed:

- latest date by which the EN has to be implemented
at national level by publication of an identical
national standard or by endorsement (dop) 2002-07-01
- latest date by which the national standards conflicting
with the EN have to be withdrawn (dow) 2004-07-01

Endorsement notice

The text of the International Standard IEC 62014-1:2001 was approved by CENELEC as a European Standard without any modification.

CONTENTS

	Page
INTRODUCTION	4
Scope and object	6
Section 1: General introduction	7
Section 2: Statement of intent	8
Section 3: General syntax rules and guidelines	10
Section 4: File header information	12
Section 5: Component description.....	14
Section 6: Model statement.....	23
Section 6a: Add submodel description	49
Section 7: Package modeling	59
Section 8: Electrical board description.....	71
Section 9: Notes on data derivation method	79

INTRODUCTION

Background of IBIS

IBIS was first developed at Intel Corporation and has been expanded to its current form (Version 3.2) through the cooperative efforts of additional analog simulator vendors, computer manufacturers, IC vendors, commercial users, and universities. In May 1993, the group formed itself into the IBIS Open Forum, an open, voluntary, cooperative association. In March 1995, the group affiliated with the EIA (now the Electronic Industries Alliance) as the EIA IBIS Open Forum.

The Forum has been and continues to meet via teleconference approximately every third week to propose updates to the IBIS standard, to help new participants, and to advance the standard. The Forum also meets in person about four times a year to exchange ideas and conduct official business.

Most of the Forum activities are handled through e-mail discussions using a reflector "ibis@eda.org." A users group reflector "ibis-users@eda.org" is also supported for users of IBIS. One can get more information on subscribing to the reflectors and to other ongoing Forum activities through the official web page: "<http://www.eigroup.org/ibis/ibis.htm>". The process of making changes and improvements to IBIS is through a "BIRD" (Buffer Issue Resolution

Document) process involving approval by the Forum voting members. Over the years the Forum has grown to over thirty voting members (requiring a modest yearly fee for administrative support), but the Forum also maintains an open, public communications policy and welcomes all interested participants regardless of membership status.

Through official EIA and ANSI (American National Standards Institution) public letter ballot processes, IBIS Version 2.1 was ratified as ANSI/EIA-656 in December 13, 1995. Version 1.1 of IBIS focused on TTL and CMOS logic components. Although never officially ratified as a national standard, IBIS Version 1.1 served as a basis for advances in Version 2.1 to increase its accuracy and number of device types that are supported.

Version 2.1 contains the following advances:

- Controlled slew rate devices
- ECL and PECL technologies
- Independent control over power rails so RS232 and other types of devices with multiple rails can be modeled
- Differential drivers and devices
- Open-drain I/O devices such as open drain and open collector devices
- Expanded package model definitions to include coupling between pins.

The Forum also voluntarily funded a parser development activity through the sale of source code licenses of "ibischk2" and has made executables of the parser code freely and publicly available to enable IBIS model checking.

Industrial advances associated with new semiconductor topologies, package design and measurement needs kept the Forum busy proposing new capabilities, eventually leading to IBIS Version 3.2. Again through official EIA and ANSI public letter ballot processes, IBIS Version 3.2 was ratified as ANSI/EIA-656-A on September 21, 1999.

Its advances include the following:

- Series and series switch models
- Multi-stage driver capability for phased stages

- Submodel capability supporting dynamic clamps and bus hold functions for active and dynamic terminations
- More specification values for overshoot and pulse immunity
- Uncoupled packages with sections and forks
- Uncoupled advanced packages known as electrical board descriptions with sections, forks and on-board components.

IBIS Version 3.2 has complied with an original Forum objective that all subsequent versions of IBIS be backward compatible with previous versions.

The Forum funded through voluntary source code license purchases the corresponding "ibischk3" parser and has made its executables freely available.

The IBIS Standard has achieved wide spread national and international support and recognition as indicated by over 40 semiconductor vendors providing IBIS models freely from their web sites. Many more IBIS models and libraries are available from commercial vendors and directly through IC vendor sales organizations. While IBIS models can be of value in all phases of a design and analysis process, they are particularly suitable for printed circuit board design tools used in conjunction with the corresponding physical and mechanical data bases describing the boards.

Future IBIS Directions

Technology continues to advance, forcing more stringent electrical requirements and newer ways of doing things. The Forum is keeping up with such advances. However, its strategy has shifted. Up to now the Forum has been adding to the existing fixed-format IBIS document. Such a process is slow and subject to unexpected interactions with existing capability. The newer approach is to create a compatible macro-language that allows more rapid reconfiguration and response to changing needs.

While the Forum has not yet ratified any of these approaches, it is pursuing these projects:

- A macro-language that fully supports IBIS Version 3.2 but can also support more advanced features and nodal component structures
- A separate Connector Specification with advanced coupled stages to support key component used to connect printed circuit boards (and possibly be used for more advanced package models)
- Some further advances in specification details beyond IBIS Version 3.2.

These projects advance the capability of IBIS in a manner that supports the existing IBIS Version 3.2 functionality, but also allows for much more rapid implementation of new requirements.

References

ANSI/EIA-656: IBIS Version 2.1 released December 13, 1995

ANSI/EIA-656-A: IBIS Version 3.2 released September 21, 1999

ELECTRONIC DESIGN AUTOMATION LIBRARIES –

Part 1: Input/output buffer information specifications (IBIS version 3.2)

Scope and object

This standard gives specifications for electronic behavioral of digital integrated circuit input/ output analog characteristics. It specifies a consistent software-parsable format for essential behavioral information.

The goal of this standard is to support all simulators of all degrees of sophistication.

Section 1

GENERAL INTRODUCTION

=====

This section gives a general overview of the remainder of this document.

Sections 2 and 3 contain general information about the IBIS versions and the general rules and guidelines. Several progressions of IBIS documents are referenced in Section 2 and in the discussion below. They are IBIS Version 1.1 (ratified August, 1993), IBIS Version 2.1 (ratified as ANSI/EIA-656 in December, 1995), and IBIS Version 3.2 (this document ratified in August, 1999).

The functionality of IBIS follows in Sections 4 through 8. Sections 4 through 6 describe the format of the core functionality of IBIS Version 1.1 and the extensions in later versions. The data in these sections are contained in .ibs files. Section 7 describes the package model format of IBIS Version 2.1 and a subsequent extension. Package models can be formatted within .ibs files or can be formatted (along with the Section 4 file header keywords) as .pkg files. Section 8 contains the Electrical Board Description format of IBIS Version 3.2. Along with Section 4 header information, electrical board descriptions must be described in separate .ebd files.

Section 9 contains some notes regarding the extraction conditions and data requirements for IBIS files. This section focuses on implementation conditions based on measurement or simulation for gathering the IBIS compliant data.

=====

Section 2

S T A T E M E N T O F I N T E N T

In order to enable an industry standard method to electronically transport IBIS Modeling Data between semiconductor vendors, simulation vendors, and end customers, this template is proposed. The intention of this template is to specify a consistent format that can be parsed by software, allowing simulation vendors to derive models compatible with their own products.

One goal of this template is to represent the current state of IBIS data, while allowing a growth path to more complex models / methods (when deemed appropriate). This would be accomplished by a revision of the base template, and possibly the addition of new keywords or categories.

Another goal of this template is to ensure that it is simple enough for semiconductor vendors and customers to use and modify, while ensuring that it is rigid enough for simulation vendors to write reliable parsers.

Finally, this template is meant to contain a complete description of the I/O elements on an entire component. Consequently, several models will need to be defined in each file, as well as a table that equates the appropriate buffer to the correct pin and signal name.

Version 3.2 of this electronic template was finalized by an industry-wide group of experts representing various companies and interests. Regular "EIA IBIS Open Forum" meetings were held to accomplish this task.

Commitment to Backward Compatibility. Version 1.0 is the first valid IBIS ASCII file format. It represents the minimum amount of I/O buffer information required to create an accurate IBIS model of common CMOS and bipolar I/O structures. Future revisions of the ASCII file will add items considered to be "enhancements" to Version 1.0 to allow accurate modeling of new, or other I/O buffer structures. Consequently, all future revisions will be considered supersets of Version 1.0, allowing backward compatibility. In addition, as modeling platforms develop support for revisions of the IBIS ASCII template, all previous revisions of the template must also be supported.

Version 1.1 update. The file "ver1_1.ibs" is conceptually the same as the 1.0 version of the IBIS ASCII format (ver1_0.ibs). However, various comments have been added for further clarification.

Version 2.0 update. The file "ver2_0.ibs" maintains backward compatibility with Versions 1.0 and 1.1. All new keywords and elements added in Version 2.0 are optional. A complete list of changes to the specification is in the IBIS Version 2.0 Release Notes document ("ver2_0.rn").

Version 2.1 update. The file "ver2_1.ibs" contains clarification text changes, corrections, and two additional waveform parameters beyond Version 2.0.

| Version 3.0 update. The file "ver3_0.ibs" adds a number of new keywords
| and functionality. A complete list of functions can be found on eda.org
| under /pub/ibis/birds/birddir.txt showing the approved Buffer Issue
| Resolution Documents (BIRDS) that have been approved for Version 3.0.

| Version 3.1 update. The file "ver3_1.ibs" contains a major reformatting of
| the document and a simplification of the wording. It also contains some
| new technical enhancements that were unresolved when Version 3.0 was
| approved.

| Version 3.2 update. The file "ver3_2.ibs" adds more technical advances and
| also a number of editorial changes documented in 12 BIRDS and also in
| responses to public letter ballot comments.

|=====

Section 3

GENERAL SYNTAX RULES AND GUIDELINES

This section contains general syntax rules and guidelines for ASCII IBIS files:

- 1) The content of the files is case sensitive, except for reserved words and keywords.
- 2) The following words are reserved words and must not be used for any other purposes in the document:
 - POWER - reserved model name, used with power supply pins,
 - GND - reserved model name, used with ground pins,
 - NC - reserved model name, used with no-connect pins,
 - NA - used where data not available.
- 3) To facilitate portability between operating systems, file names used in the IBIS file must only have lower case characters. File names should have a basename of no more than twenty characters followed by a period ('.') , followed by a file name extension of no more than three characters. The file name and extension must use characters from the set (space, ' ', 0x20 is not included):

```
a b c d e f g h i j k l m n o p q r s t u v w x y z
0 1 2 3 4 5 6 7 8 9 _ ^ $ ~ ! # % & - { } ) ( @ ' `
```

The file name and extension are recommended to be lower case on systems that support such names.

- 4) A line of the file may have at most 80 characters, followed by a line termination sequence. The line termination sequence must be one of the following two sequences: a linefeed character, or a carriage return followed by linefeed character.
- 5) Anything following the comment character is ignored and considered a comment on that line. The default "|" (pipe) character can be changed by the keyword [Comment Char] to any other character. The [Comment Char] keyword can be used throughout the file as desired.
- 6) Keywords must be enclosed in square brackets, [], and must start in column 1 of the line. No space or tab is allowed immediately after the opening bracket '[' or immediately before the closing bracket ']'. If used, only one space (' ') or underscore ('_') character separates the parts of a multi-word keyword.
- 7) Underscores and spaces are equivalent in keywords. Spaces are not allowed in subparameter names.

8) Valid scaling factors are:

T = tera	k = kilo	n = nano
G = giga	m = milli	p = pico
M = mega	u = micro	f = femto

When no scaling factors are specified, the appropriate base units are assumed. (These are volts, amperes, ohms, farads, henries, and seconds.) The parser looks at only one alphabetic character after a numerical entry, therefore it is enough to use only the prefixes to scale the parameters. However, for clarity, it is allowed to use full abbreviations for the units, (e.g., pF, nH, mA, mOhm). In addition, scientific notation IS allowed (e.g., 1.2345e-12).

- 9) The I-V data tables should use enough data points around sharply curved areas of the I-V curves to describe the curvature accurately. In linear regions there is no need to define unnecessary data points.
- 10) The use of tab characters is legal, but they should be avoided as much as possible. This is to eliminate possible complications that might arise in situations when tab characters are automatically converted to multiple spaces by text editing, file transferring and similar software. In cases like that, lines might become longer than 80 characters, which is illegal in IBIS files.
- 11) Currents are considered positive when their direction is into the component.
- 12) All temperatures are represented in degrees Celsius.
- 13) Important supplemental information is contained in the last section, "NOTES ON DATA DERIVATION METHOD", concerning how data values are derived.
- 14) Only ASCII characters, as defined in ANSI Standard X3.4-1986, may be used in an IBIS file. The use of characters with codes greater than hexadecimal 07E is not allowed. Also, ASCII control characters (those numerically less than hexadecimal 20) are not allowed, except for tabs or in a line termination sequence. As mentioned in item 10 above, the use of tab characters is discouraged.
- =====

Section 4

FILE HEADER INFORMATION

Keyword:	[IBIS Ver]
Required:	Yes
Description:	Specifies the IBIS template version. This keyword informs electronic parsers of the kinds of data types that are present in the file.
Usage Rules:	[IBIS Ver] must be the first keyword in any IBIS file. It is normally on the first line of the file, but can be preceded by comment lines that must begin with a " ".
[IBIS Ver]	3.2 Used for template variations
Keyword:	[Comment Char]
Required:	No
Description:	Defines a new comment character to replace the default " " (pipe) character, if desired.
Usage Rules:	The new comment character to be defined must be followed by the underscore character and the letters "char". For example: " _char" redundantly redefines the comment character to be the pipe character. The new comment character is in effect only following the [Comment Char] keyword. The following characters MAY be used:
	! " # \$ % & ' () * , : ; < > ? @ \ ^ ` { } ~
Other Notes:	The [Comment Char] keyword can be used throughout the file, as desired.
[Comment Char]	_char
Keyword:	[File Name]
Required:	Yes
Description:	Specifies the name of the IBIS file.
Usage Rules:	The file name must conform to the rules in paragraph 3 of Section 3, "GENERAL SYNTAX RULES AND GUIDELINES". In addition, the file name must use the extension ".ibs", ".pkg", or ".ebd". The file name must be the actual name of the file.
[File Name]	ver3_2.ibs
Keyword:	[File Rev]
Required:	Yes
Description:	Tracks the revision level of a particular .ibs file.
Usage Rules:	Revision level is set at the discretion of the engineer defining the file. The following guidelines are recommended:

	0.x	silicon and file in development	
	1.x	pre-silicon file data from silicon model only	
	2.x	file correlated to actual silicon measurements	
	3.x	mature product, no more changes likely	

[File Rev]	1.0	Used for .ibs file variations	
=====			
	Keywords:	[Date], [Source], [Notes], [Disclaimer], [Copyright]	
	Required:	No	
	Description:	Optionally clarifies the file.	
	Usage Rules:	The keyword arguments can contain blanks, and be of any format. The [Date] keyword argument is limited to a maximum of 40 characters, and the month should be spelled out for clarity.	
		Because IBIS model writers may consider the information in these keywords essential to users, and sometimes legally required, design automation tools should make this information available. Derivative models should include this text verbatim. Any text following the [Copyright] keyword must be included in any derivative models verbatim.	

[Date]	August 20, 1999	The latest file revision date	
[Source]	Put originator and the source of information here. For example:		
	From silicon level SPICE model at Intel.		
	From lab measurement at IEI.		
	Compiled from manufacturer's data book at Quad Design, etc.		
[Notes]	Use this section for any special notes related to the file.		
[Disclaimer]	This information is for modeling purposes only, and is not guaranteed. May vary by component		
[Copyright]	Copyright 1999, XYZ Corp., All Rights Reserved		
=====			

Section 5

C O M P O N E N T D E S C R I P T I O N

Keyword: [Component]
Required: Yes
Description: Marks the beginning of the IBIS description of the integrated circuit named after the keyword.
Sub-Params: Si_location, Timing_location
Usage Rules: If the .ibs file contains data for more than one component, each section must begin with a new [Component] keyword. The length of the component name must not exceed 40 characters, and blank characters are allowed.

NOTE: Blank characters are not recommended due to usability issues.

Si_location and Timing_location are optional and specify where the Signal Integrity and Timing measurements are made for the component. Allowed values for either subparameter are 'Die' or 'Pin'. The default location is at the 'Pin'.

[Component] 7403398 MC452

Si_location	Pin		Optional subparameters to give measurement
Timing_location	Die		location positions

Keyword: [Manufacturer]
Required: Yes
Description: Specifies the manufacturer's name of the component.
Usage Rules: The length of the manufacturer's name must not exceed 40 characters (blank characters are allowed, e.g., Texas Instruments). In addition, each manufacturer must use a consistent name in all .ibs files.

[Manufacturer] Intel Corp.

Keyword: [Package]
Required: Yes
Description: Defines a range of values for the default packaging resistance, inductance, and capacitance of the component pins.
Sub-Params: R_pkg, L_pkg, C_pkg
Usage Rules: The typical (typ) column must be specified. If data for the other columns are not available, they must be noted with "NA".
Other Notes: If RLC parameters are available for individual pins, they can be listed in columns 4-6 under keyword [Pin]. The values listed in the [Pin] description section override the default values defined here. Use the [Package Model] keyword for more complex package descriptions. If defined, the [Package Model] data overrides the values in the [Package] keyword.

Regardless, the data listed under the [Package] keyword must still contain valid data.

[Package]

variable	typ	min	max
R_pkg	250.0m	225.0m	275.0m
L_pkg	15.0nH	12.0nH	18.0nH
C_pkg	18.0pF	15.0pF	20.0pF

Keyword: [Pin]

Required: Yes

Description: Associates the component's I/O models to its various external pin names and signal names.

Sub-Params: signal_name, model_name, R_pin, L_pin, C_pin

Usage Rules: All pins on a component must be specified. The first column must contain the pin name. The second column, signal_name, gives the data book name for the signal on that pin. The third column, model_name, maps a pin to a specific I/O buffer model or model selector name. Each model_name must have a corresponding model or model selector name listed in a [Model] or [Model Selector] keyword below, unless it is a reserved model name (POWER, GND, or NC).

Each line must contain either three or six columns. A pin line with three columns only associates the pin's signal and model. Six columns can be used to override the default package values (specified under [Package]) FOR THAT PIN ONLY. When using six columns, the headers R_pin, L_pin, and C_pin must be listed. If "NA" is in columns 4 through 6, the default packaging values must be used. The headers R_pin, L_pin, and C_pin may be listed in any order.

Column length limits are:

[Pin]	5 characters max
model_name	20 characters max
signal_name	20 characters max
R_pin	9 characters max
L_pin	9 characters max
C_pin	9 characters max

[Pin]	signal_name	model_name	R_pin	L_pin	C_pin
1	RAS0#	Buffer1	200.0m	5.0nH	2.0pF
2	RAS1#	Buffer2	209.0m	NA	2.5pF
3	EN1#	Input1	NA	6.3nH	NA
4	A0	3-state			
5	D0	I/O1			
6	RD#	Input2	310.0m	3.0nH	2.0pF
7	WR#	Input2			
8	A1	I/O2			
9	D1	I/O2			
10	GND	GND	297.0m	6.7nH	3.4pF
11	RDY#	Input2			
12	GND	GND	270.0m	5.3nH	4.0pF

.
.

18	Vcc3	POWER			
19	NC	NC			
20	Vcc5	POWER	226.0m	NA	1.0pF

Keyword: [Package Model]

Required: No

Description: Indicates the name of the package model to be used for the component

Usage Rules: The package model name is limited to 40 characters. Spaces are allowed in the name. The name should include the company name or initials to help ensure uniqueness. The simulator will search for a matching package model name as an argument to a [Define Package Model] keyword in the current IBIS file first. If a match is not found, the simulator will next look for a match in an external .pkg file. If the matching package model is in an external .pkg file, it must be located in the same directory as the .ibs file. The file names of .pkg files must follow the rules for file names given in section 3, General Syntax Rules and Guidelines.

Other Notes: Use the [Package Model] keyword within a [Component] to indicate which package model should be used for that component. The specification permits .ibs files to contain [Define Package Model] keywords as well. These are described in the "Package Modeling" section near the end of this specification. When package model definitions occur within a .ibs file, their scope is "local"--they are known only within that .ibs file and no other. In addition, within that .ibs file, they override any globally defined package models that have the same name.

[Package Model] QS-SMT-cer-8-pin-pkgs

Keyword: [Pin Mapping]

Required: No

Description: Used to indicate which power and ground buses a given driver, receiver, or terminator is connected to.

Sub-Params: pulldown_ref, pullup_ref, gnd_clamp_ref, power_clamp_ref

Usage Rules: Each power and ground bus is given a unique name that must not exceed 15 characters. The first column contains a pin name. Each pin name must match one of the pin names declared previously in the [Pin] section of the IBIS file. The second column, pulldown_ref, designates the ground bus connections for that pin. Here the term ground bus can also mean another power bus. The third column pullup_ref designates the power bus connection. The fourth and fifth columns gnd_clamp_ref and power_clamp_ref contain entries, if needed, to specify different ground bus and power bus connections than those previously specified.

If the [Pin Mapping] keyword is present, then the bus connections for EVERY pin listed in the [Pin] section must be given.

Each line must contain either three or five columns. Use the

NC reserved word for entries that are not needed or that follow the conditions below:

All entries with identical labels are assumed to be connected. Each unique entry label must connect to at least one pin whose model_name is POWER or GND.

If a pin has no connection, then both the pulldown_ref and pullup_ref subparameters for it will be NC.

GND and POWER pin entries and buses are designated by entries in either the pulldown_ref or pullup_ref columns. There is no implied association to any column other than through explicit designations in other pins.

For any other type of pin, the pulldown_ref column contains the power connection for the [Pulldown] table for non-ECL type [Model]s. This is also the power connection for the [GND Clamp] table and the [Rgnd] model unless overridden by a specification in the gnd_clamp_ref column.

Also, the pullup_ref column contains the power connection for the [Pullup] table and, for ECL type models, the [Pulldown] table. This is also the power connection for the [POWER Clamp] table and the [Rpower] model unless overridden by a specification in the power_clamp_ref column.

The column length limits are:

[Pin Mapping]	5 characters max
pulldown_ref	15 characters max
pullup_ref	15 characters max
gnd_clamp_ref	15 characters max
power_clamp_ref	15 characters max

When 5 columns are specified, the headings gnd_clamp_ref and power_clamp_ref must be used. Otherwise, these headings can be omitted.

[Pin Mapping]	pulldown_ref	pullup_ref	gnd_clamp_ref	power_clamp_ref
1	GNDBUS1	PWRBUS1	Signal pins and their associated ground and power connections	
2	GNDBUS2	PWRBUS2		
3	GNDBUS1	PWRBUS1	GNDCLMP	PWRCLAMP
4	GNDBUS2	PWRBUS2	GNDCLMP	PWRCLAMP
5	GNDBUS2	PWRBUS2	NC	PWRCLAMP
6	GNDBUS2	PWRBUS2	GNDCLMP	NC
			Some possible clamping connections are shown above for illustration purposes	
11	GNDBUS1	NC	One set of ground connections. NC indicates no connection to power bus.	
12	GNDBUS1	NC		
13	GNDBUS1	NC		
21	GNDBUS2	NC	Second set of ground connections	
22	GNDBUS2	NC		
23	GNDBUS2	NC		

31	NC	PWRBUS1	One set of power connections.
32	NC	PWRBUS1	NC indicates no connection to
33	NC	PWRBUS1	ground bus.
41	NC	PWRBUS2	Second set of power connections
42	NC	PWRBUS2	
43	NC	PWRBUS2	
51	GNDCLMP	NC	Additional power connections
52	NC	PWRCLMP	for clamps

=====

Keyword: [Diff Pin]

Required: No

Description: Associates differential pins, their differential threshold voltages, and differential timing delays.

Sub-Params: inv_pin, vdiff, tdelay_typ, tdelay_min, tdelay_max

Usage Rules: Enter only differential pin pairs. The first column, [Diff Pin], contains a non-inverting pin name. The second column, inv_pin, contains the corresponding inverting pin name for I/O output. Each pin name must match the pin names declared previously in the [Pin] section of the IBIS file. The third column, vdiff, contains the specified output and differential threshold voltage between pins if the pins are Input or I/O model types. For output only differential pins, the vdiff entry is 0 V. The fourth, fifth, and sixth columns, tdelay_typ, tdelay_min, and tdelay_max, contain launch delays of the non-inverting pins relative to the inverting pins. The values can be of either polarity.

If a pin is a differential input pin, the differential input threshold (vdiff) overrides and supersedes the need for Vinh and Vinl.

If vdiff is not defined for a pin that is defined as requiring a Vinh by its [Model] type, vdiff is set to the default value of 200 mV.

Other Notes: The output pin polarity specification in the table overrides the [Model] Polarity specification such that the pin in the [Diff Pin] column is Non-Inverting and the pin in the inv_pin column is Inverting. This convention enables one [Model] to be used for both pins.

Column length limits are:

[Diff Pin]	5 characters max
inv_pin	5 characters max
vdiff	9 characters max
tdelay_typ	9 characters max
tdelay_min	9 characters max
tdelay_max	9 characters max

Each line must contain either four or six columns. If "NA" is entered in the vdiff, tdelay_typ, or tdelay_min columns, its entry is interpreted as 0 V or 0 ns. If "NA" appears in the tdelay_max column, its value is interpreted as the tdelay_typ

value. When using six columns, the headers `tdelay_min` and `tdelay_max` must be listed. Entries for the `tdelay_min` column are based on minimum magnitudes; and `tdelay_max` column, maximum magnitudes. One entry of `vdiff`, regardless of its polarity, is used for difference magnitudes.

```
-----
[Diff Pin]  inv_pin  vdiff  tdelay_typ  tdelay_min  tdelay_max
-----
3           4       150mV  -1ns       0ns        -2ns | Input or I/O pair
7           8        0V     1ns       NA         NA  | Output* pin pair
9          10        NA     NA        NA         NA  | Output* pin pair
16          15      200mV   1ns       | Input or I/O pin pair
20          19        0V     NA       | Output* pin pair, tdelay = 0
22          21        NA     NA       | Output*, tdelay = 0
                                     | * Could be Input or I/O with vdiff = 0
-----
```

Keyword: [Series Pin Mapping]

Required: No

Description: Used to associate two pins joined by a series model.

Sub-Params: `pin_2`, `model_name`, `function_table_group`

Usage Rules: Enter only series pin pairs. The first column, [Series Pin Mapping], contains the series pin for which input impedances are measured. The second column, `pin_2`, contains the other connection of the series model. Each pin must match the pin names declared previously in the [Pin] section of the IBIS file. The third column, `model_name`, associates the Series or Series_switch model for the pair of pins in the first two columns. The fourth column, `function_table_group`, contains an alphanumeric designator string to associate those sets of Series_switch pins that are switched together.

Each line must contain either three or four columns. When using four columns, the header `function_table_group` must be listed.

One possible application is to model crossbar switches where the straight through On paths are indicated by one designator and the cross over On paths are indicated by another designator. If the model referenced is a Series model, then the `function_table_group` entry is omitted.

Column length limits are:

[Series Pin Mapping]	5 characters max
<code>pin_2</code>	5 characters max
<code>model_name</code>	20 characters max
<code>function_table_group</code>	20 characters max

Other Notes: If the `model_name` is for a non-symmetrical series model, then the order of the pins is important. The [Series Pin Mapping] and `pin_2` entries must be in the columns that correspond with Pin 1 and Pin 2 of the referenced model.

This mapping covers only the series paths between pins. The package parasitics and any other elements such as additional capacitance or clamping circuitry are defined by the `model_name` that is referenced in the [Pin] keyword. The

model_names under the [Pin] keyword that are also referenced by the [Series Pin Mapping] keyword may include any legal model or reserved model except for Series and Series_switch models. Normally the pins will reference a [Model] whose Model_type is 'Terminator'. For example, a Series_switch model may contain Terminator models on EACH of the pins to describe both the capacitance on each pin and some clamping circuitry that may exist on each pin. In a similar manner, Input, I/O or Output models may exist on each pin of a Series model that is serving as a differential termination.

[Series Pin Mapping]

pin_2	model_name	function_table_group
2	3	CBTSeries
5	6	CBTSeries
9	8	CBTSeries
12	11	CBTSeries
22	23	CBTSeries
25	26	CBTSeries
22	26	CBTSeries
25	23	CBTSeries
32	33	Fixed_series

1

2

3

4

5

5

6

6

| Four independent groups

| Straight through path

| Cross over path

| No group needed

Keyword:

[Series Switch Groups]

Required:

No

Description:

Used to define allowable switching combinations of series switches described using the names of the groups in the [Series Pin Mapping] keyword function_table_group column

Sub-Params:

On, Off

Usage Rules:

Each state line contains an allowable configuration. A typical state line will start with 'On' followed by all of the on-state group names or an 'Off' followed by all of the off-state group names. Only one of 'On' or 'Off' is required since the undefined states are presumed to be opposite of the explicitly defined states. The state line is terminated with the slash '/', even if it extends over several lines to fit within the 80 character column width restriction.

The group names in the function_table_group are used to associate switches whose switching action is synchronized by a common control function. The first line defines the assumed (default) state of the set of series switches. Other sets of states are listed and can be selected through a user interface or through automatic control.

[Series Switch Groups]

Function Group States

On 1 2 3 4 /

Off 1 2 3 4 /

On 1 /

On 2 /

On 3 /

| Default setting is all switched On.

| All Off setting.

| Other possible combinations below.

```
On 4 /
On 1 2 /
On 1 3 /
On 1 4 /
On 2 3 /
On 2 4 /
On 3 4 /
On 1 2 3 /
On 1 2 4 /
On 1 3 4 /
On 2 3 4 /
| Off 4 /           | The last four lines above could have been replaced
| Off 3 /           | with these four lines with the same meaning.
| Off 2 /
| Off 1 /
|
On 5 /           | Crossbar switch straight through connection
On 6 /           | Crossbar cross over connection
Off 5 6 /        | Crossbar open switches
```

```
=====
| Keyword: [Model Selector]
| Required: No.
| Description: Used to pick a [Model] from a list of [Model]s for a pin which
|                 uses a programmable buffer.
| Usage Rules: A programmable buffer must have an individual [Model] section
|                 for each one of its modes used in the .ibs file. The names of
|                 these [Model]s must be unique and can be listed under the
|                 [Model Selector] keyword and/or pin list. The name of the
|                 [Model Selector] keyword must match the corresponding model
|                 name listed under the [Pin] or [Series Pin Mapping] keyword
|                 and must not contain more than 20 characters. A .ibs file
|                 must contain enough [Model Selector] keywords to cover all of
|                 the model selector names specified under the [Pin] and [Series
|                 Pin Mapping] keywords.
|
|                 The section under the [Model Selector] keyword must have two
|                 fields. The two fields must be separated by at least one
|                 white space. The first field lists the [Model] name (up to 20
|                 characters long). The second field contains a short
|                 description of the [Model] shown in the first field. The
|                 contents and format of this description is not standardized,
|                 however it shall be limited in length so that none of the
|                 descriptions exceed the 80-character length of the line that
|                 it started on. The purpose of the descriptions is to aid the
|                 user of the simulator tool in making intelligent buffer mode
|                 selections and it can be used by the simulator tool in a user
|                 interface dialog box as the basis of an interactive buffer
|                 selection mechanism.
|
|                 The first entry under the [Model Selector] keyword shall be
|                 considered the default by the simulator tool for all those
|                 pins which call this [Model Selector].
|
|                 The operation of this selection mechanism implies that a group
|                 of pins which use the same programmable buffer (i.e. model
|                 selector name) will be switched together from one [Model] to
```

another. Therefore, if two groups of pins, for example an address bus and a data bus, use the same programmable buffer, and the user must have the capability to configure them independently, one can use two [Model Selector] keywords with unique names and the same list of [Model] keywords; however, the usage of the [Model Selector] is not limited to these examples. Many other combinations are possible.

[Pin]	signal_name	model_name	R_pin	L_pin	C_pin
1	RAS0#	Progbuffer1	200.0m	5.0nH	2.0pF
2	EN1#	Input1	NA	6.3nH	NA
3	A0	3-state			
4	D0	Progbuffer2			
5	D1	Progbuffer2	320.0m	3.1nH	2.2pF
6	D2	Progbuffer2			
7	RD#	Input2	310.0m	3.0nH	2.0pF

.					
.					
.					
18	Vcc3	POWER			

[Model Selector] Progbuffer1

OUT_2	2 mA buffer without slew rate control
OUT_4	4 mA buffer without slew rate control
OUT_6	6 mA buffer without slew rate control
OUT_4S	4 mA buffer with slew rate control
OUT_6S	6 mA buffer with slew rate control

[Model Selector] Progbuffer2

OUT_2	2 mA buffer without slew rate control
OUT_6	6 mA buffer without slew rate control
OUT_6S	6 mA buffer with slew rate control
OUT_8S	8 mA buffer with slew rate control
OUT_10S	10 mA buffer with slew rate control

=====

Section 6

M O D E L S T A T E M E N T

Keyword: [Model]

Required: Yes

Description: Used to define a model, and its attributes.

Sub-Params: Model_type, Polarity, Enable, Vinl, Vinh, C_comp, Vmeas, Cref, Rref, Vref

Usage Rules: Each model type must begin with the keyword [Model]. The model name must match the one that is listed under a [Pin], [Model Selector] or [Series Pin Mapping] keyword and must not contain more than 20 characters. A .ibs file must contain enough [Model] keywords to cover all of the model names specified under the [Pin], [Model Selector] and [Series Pin Mapping] keywords, except for those model names that use reserved words (POWER, GND and NC).

Model_type must be one of the following:

Input, Output, I/O, 3-state, Open_drain, I/O_open_drain, Open_sink, I/O_open_sink, Open_source, I/O_open_source, Input_ECL, Output_ECL, I/O_ECL, 3-state_ECL, Terminator, Series, and Series_switch.

Special usage rules apply to the following. Some definitions are included for clarification:

Input	These model types must have Vinl and Vinh defined. If they are not defined, the parser issues a warning and the default values of Vinl = 0.8 V and Vinh = 2.0 V are assumed.
I/O	
I/O_open_drain	
I/O_open_sink	
I/O_open_source	

Input_ECL	These model types must have Vinl and Vinh defined. If they are not defined, the parser issues a warning and the default values of Vinl = -1.475 V and Vinh = -1.165 V are assumed.
I/O_ECL	

Terminator	This model type is an input-only model that can have analog loading effects on the circuit being simulated but has no digital logic thresholds. Examples of Terminators are: capacitors, termination diodes, and pullup resistors.
------------	--

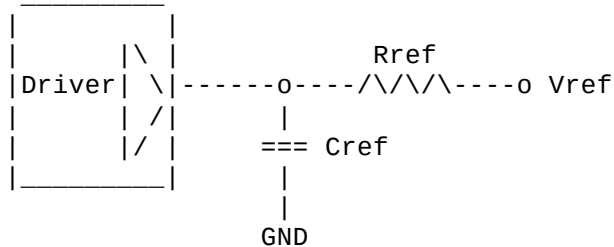
Output	This model type indicates that an output always sources and/or sinks current and cannot be disabled.
--------	--

3-state	This model type indicates that an output
---------	--

	can be disabled, i.e. put into a high impedance state.
Open_sink Open_drain	These model types indicate that the output has an OPEN side (do not use the [Pullup] keyword, or if it must be used, set I = 0 mA for all voltages specified) and the output SINKS current. Open_drain model type is retained for backward compatibility.
Open_source	This model type indicates that the output has an OPEN side (do not use the [Pulldown] keyword, or if it must be used, set I = 0 mA for all voltages specified) and the output SOURCES current.
Input_ECL Output_ECL I/O_ECL 3-state_ECL	These model types specify that the model represents an ECL type logic that follows different conventions for the [Pulldown] keyword.
Series	This model type is for series models that can be described by [R Series], [L Series], [Rl Series], [C Series], [Lc Series], [Rc Series], [Series Current] and [Series MOSFET] keywords
Series_switch	This model type is for series switch models that can be described by [On], [Off], [R Series], [L Series], [Rl Series], [C Series], [Lc Series], [Rc Series], [Series Current] and [Series MOSFET] keywords

The Model_type and C_comp subparameters are required. The Polarity, Enable, Vinl, Vinh, Vmeas, Cref, Rref, and Vref subparameters are optional. C_comp defines the silicon die capacitance. This value should not include the capacitance of the package. C_comp is allowed to use "NA" for the min and max values only. The Polarity subparameter can be defined as either Non-Inverting or Inverting, and the Enable subparameter can be defined as either Active-High or Active-Low.

The Cref and Rref subparameters correspond to the test load that the semiconductor vendor uses when specifying the propagation delay and/or output switching time of the model. The Vmeas subparameter is the reference voltage level that the semiconductor vendor uses for the model. Include Cref, Rref, Vref, and Vmeas information to facilitate board-level timing simulation. The assumed connections for Cref, Rref, and Vref are shown in the following diagram:



Other Notes: A complete [Model] description normally contains the following keywords: [Voltage Range], [Pullup], [Pulldown], [GND Clamp], [POWER Clamp], and [Ramp]. A Terminator model uses one or more of the [Rgnd], [Rpower], [Rac], and [Cac]. However, some models may have only a subset of these keywords. For example, an input structure normally only needs the [Voltage Range], [GND Clamp], and possibly the [POWER Clamp] keywords. If one or more of [Rgnd], [Rpower], [Rac], and [Cac] keywords are used, then the Model_type must be Terminator.

Signals	CLK1, CLK2,...	Optional signal list, if desired	
[Model]	Clockbuffer		
Model_type	I/O		
Polarity	Non-Inverting		
Enable	Active-High		
Vinl = 0.8V		input logic "low" DC voltage, if any	
Vinh = 2.0V		input logic "high" DC voltage, if any	
Vmeas = 1.5V		Reference voltage for timing measurements	
Cref = 50pF		Timing specification test load capacitance value	
Rref = 500		Timing specification test load resistance value	
Vref = 0		Timing specification test load voltage	
variable	typ	min	max
C_comp	12.0pF	10.0pF	15.0pF

Keyword: [Model Spec]
Required: No
Sub-Params: Vinh, Vinl, Vinh+, Vinh-, Vinl+, Vinl-, S_overshoot_high, S_overshoot_low, D_overshoot_high, D_overshoot_low, D_overshoot_time, Pulse_high, Pulse_low, Pulse_time, Vmeas
Description: The [Model Spec] keyword defines four columns under which specification subparameters are defined.

The following subparameters are defined:

Vinh	Input voltage threshold high
Vinl	Input voltage threshold low
Vinh+	Hysteresis threshold high max Vt+
Vinh-	Hysteresis threshold high min Vt+
Vinl+	Hysteresis threshold low max Vt-
Vinl-	Hysteresis threshold low min Vt-
S_overshoot_high	Static overshoot high voltage
S_overshoot_low	Static overshoot low voltage
D_overshoot_high	Dynamic overshoot high voltage
D_overshoot_low	Dynamic overshoot low voltage
D_overshoot_time	Dynamic overshoot time
Pulse_high	Pulse immunity high voltage
Pulse_low	Pulse immunity low voltage